## 题目查重算法介绍

项目背景：对于导出的题目数据，分析其中存在的相同及相似题目，进而实现去除重复题目的功能。

项目主要分两部分：包括对数据的文本预处理及对题目的相似度评估分析。

1. 文本预处理
2. 对存储在raw\_data的原始题库数据文件进行处理，针对不同导出的语文及数学又不同的处理方式，主要实现解析其中的html内容，获取文本描述的题目信息，生成的文件保存在data\_source中。

通过src/preprocess/phrase\_chinese\_data.py处理语文数据。

通过src/preprocess/phrase\_math\_data.py处理数学数据。

1. 将data\_source中的数据进一步处理，通过src/preprocess/split.py区分处理不同题目情况，当前暂不考虑套题等情况。结果保存在data\_split中对应表格文件的“正常”sheet中。
2. 将data\_split中的数据分别通过preprocess文件进行预处理，去除不合法字符及无意义字符，按照题目类型存放在不同的csv文件中，生成的结果存放在data/preprocess\_result\_xxx/中。
3. 相似度分析

经过实验比较发现，基于编辑距离的Levenshtein\_ratio对题目是否相似的评估效果最好，因此采用该方法比较两个问题文本描述的相似度。

在效果比较中，分别针对不同类型的题目做不同的处理，将同一类型的问题描述语句两两比较，计算Levenshtein\_ratio的评估值，并将结果从大到小倒序排序。实验发现相似度小于0.5的题目对不会相同，因此结果仅展示相似度大于0.5的题目对比较。

Levenshtein\_ratio的取值范围在0到1，结果越小表面两个题目越不相似，反之结果越接近1表明题目越相似。

对于单选题与多选题，除了比较题干的相似度，我们将选项也纳入比较范围，若题干与选项均相似，才会认为两个题目相似。其余填空，判断，主观题仅比较题干相似度。

实验结果发现，纯文本描述的语文题库的题目比较效果优于含有公式字符的数学题目比较。当前相似度计算的方式是基于字符的，对于公式等非文本情况的比较效果较差。